

LAB - 1 - 08/04/24

## PROGRAM 1: Quadratic

```
import java.util.Scanner;
```

```
import java.lang.Math;
```

```
class Quadratic
```

{

```
public static void main(String[] args)
```

{

```
    int a, b, c;
```

```
    System.out.println("Enter the values of a, b, c");
```

```
    Scanner s1 = new Scanner(System.in);
```

```
    a = s1.nextInt();
```

```
    b = s1.nextInt();
```

```
    c = s1.nextInt();
```

```
    double d = b * b - 4 * a * c;
```

```
    System.out.println("a = " + a + " b = " + b + " c = " + c);
```

```
    if (a == 0)
```

{

~~System.out.println("The equation is not quadratic");~~

{

~~else if (d > 0)~~

{

~~System.out.println("The equation has 2 real and different  
Solutions");~~~~double x1 = (-b + Math.sqrt(d)) / (2 \* a);~~~~double x2 = (-b - Math.sqrt(d)) / (2 \* a);~~~~System.out.println("x1 = " + x1);~~~~System.out.println("x2 = " + x2);~~

{

else if ( $d=0$ )

{

Sopln ("The equation has real and equal solutions").

double  $\pi_1 = -b / (2 * a)$ ;

double  $\pi_2 = -b / (2 * a)$ ;

Sopln (" $\pi_1 = \pi_2 =$  " +  $\pi_1$ );

}

else if ( $d < 0$ )

{

Sopln ("The equation has unreal solutions");

}

}

OUTPUT:

Enter the values of  $a, b, c$  respectively :

2 4 5

The equation has unreal solutions

Enter the values of  $a, b, c$  respectively

1 -7 10

The equation has two real and different solutions

$\pi_1 = 5.0$

$\pi_2 = 2.0$

Enter the values of  $a, b, c$  respectively

1 -4 4

The equation has real and equal solutions

$\pi_1 = \pi_2 = 2.0$

## PROGRAM 2 - Student

```
import java.util.Scanner;
```

```
class Scanner
```

```
{
```

```
String usn;
```

```
String name;
```

```
int marks[] = new int [6];
```

```
void Details()
```

```
{
```

```
Scanner s = new Scanner (System.in);
```

```
Sopln ("Enter USN");
```

```
usn = s.next();
```

```
Sopln ("Enter name");
```

```
name = s.next();
```

```
Sopln ("Enter marks for 6 subjects");
```

```
for (int i=0; i<6; i++)
```

```
{
```

```
Sopln ("Subject" + (i+1) + ":"),
```

```
marks[i] = s.nextInt();
```

```
{
```

```
double Percentage()
```

```
{
```

```
int total=0;
```

```
for (int i=0; i<6; i++)
```

```
{
```

```
total += marks[i];
```

```
{
```



PARAGRAPH 3 -

```
import java.util.Scanner;  
class Books  
{  
    String Name;  
    String Author;  
    int price;  
    int numPages;  
  
    Books (String Name, String Author, int price, int numPages)  
    {  
        this.Name = Name;  
        this.Author = Author;  
        this.price = price;  
        this.numPages = numPages;  
  
    }  
  
    public String toString()  
    {  
        String name, author, price, numPages;  
        name = "Book name :" + this.Name + "\n";  
        author = "Author name :" + this.Author + "\n";  
        numPages = "Number of pages " + this.numPages + "\n";  
        price = "Price " + this.price + "\n";  
        return name + author + numPages + price;  
    }  
}
```

08.01.24

Class Main

{

```
public static void main(String args[])
```

{

```
Scanner s = new Scanner (System.in);
```

int n;

String Name;

String Author;

int price;

int numPages;

```
s.nextLine("Enter the number of books");
```

n = s.nextInt();

```
Books b[] = new Books[n];
```

```
for (int i=0; i<n; i++)
```

{

```
s.nextLine("Enter Book Name");
```

Name = s.next();

```
s.nextLine("Enter the Author Name");
```

Author = s.next();

```
s.nextLine("Enter Price");
```

price = s.nextInt();

```
s.nextLine("Enter the number of Pages");
```

numPages = s.nextInt();

```
b[i] = new Books (Name, Author, price, numPages);
```

}

```
for (int i=0; i<n; i++)
```

{

```
s.nextLine("Books" + (i+1) + "\n");
```

```
s.nextLine(b[i].toString());
```

}

OUTPUT:

Enter the number of books:

1

Enter the name of the book

Percy Jackson

Enter the <sup>name of</sup> Author

Rick Riordan

Enter the price

1000

Enter the number of Pages

800

Book name: Percy Jackson

Author name: Rick Riordan

Price: 1000

Number of Pages: 800

Droh  
8/1/21

1. Develop a Java program to create an abstract class named `shape` that contains two integers and an empty method named `printArea()`.

### PROGRAM

```
import java.util.*  
abstract class Shape  
{  
    protected int dimension 1;  
    protected int dimension 2;  
    public shape (int dimension 1, int dimension 2)  
    {  
        this . dimension 1 = dimension 1;  
        this . dimension 2 = dimension 2;  
    }
```

```
    public abstract void printArea();  
}
```

```
class Rectangle extends Shape  
{
```

```
    public Rectangle (int length, int width)  
    {
```

```
        super (length, width);  
    }
```

```
    public void printArea()  
    {
```

```
        int area = dimension 1 * dimension 2;  
        System.out.println ("Area of Rectangle" + area);  
    }
```

```

class Triangle extends Shape
{
    public Triangle (int base, int height)
    {
        super(base, height);
    }

    public void printArea()
    {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle " + area);
    }
}

class Circle extends Shape
{
    public Circle (int radius)
    {
        super(radius, 0);
    }

    public void printArea()
    {
        double area = Math.PI * dimension1 * dimension2;
        System.out.println("Area of Circle " + area);
    }
}

public class Main
{
    public static void main (String [] args)
    {
        Rectangle r = new Rectangle(4,5);
        r.printArea();

        Triangle t = new Triangle(3,6);
        t.printArea();
    }
}

```

Circle c = new Circle(7);  
 c.printArea();

OUTPUT:

Area of Rectangle : 20  
 Area of Triangle : 9  
 Area of Circle : 153.000

200  
22/1/24

2.

11

PROGRAM

class Bank

{

public static void main (String[] args)

{

SavingsAccount savingsAccount = new SavingsAccount ("John Doe", "SA1001");

CurrentAccount currentAccount = new CurrentAccount ("Jane Smith", "CA2002");

savingsAccount.deposit(5000);

savingsAccount.displayBalance();

savingsAccount.computeInterest();

savingsAccount.displayBalance();

currentAccount.withdraw(500);

currentAccount.displayBalance();

{}

class Account

{

protected String customerName;

protected String accountNumber;

protected double balance;

public Account (String customerName, String accountNumber)

{

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = 0;

{

public void deposit (double amount)

{

balance += amount;

System.out.println("Deposit of " + amount + " Successful");

{

public void displayBalance()

{

System.out.println("Account Number : " + accountNumber + " Balance " + balance);

{

class SavingsAccount extends Account

{

private double minimumBalance = 500;

public SavingsAccount (String customerName, String accountNumber)

{

super(customerName, accountNumber);

{

public void computeInterest()

{

double interestRate = 0.05;

double interest = balance \* interestRate;

11

```

balance += interest;
System.out.println("Interest of $" + interest + " computed and added to
the balance");
}

public void withdraw(double amount)
{
    if (balance != amount)
    {
        balance -= amount;
        System.out.println("Withdrawal of $" + amount + " successful");
    }
    else
    {
        System.out.println("Insufficient funds for
withdrawal");
    }
}

class CurrentAccount extends Account
{
    private double minimumBalance = 1000;
    public CurrentAccount (String customerName, String accountNumber)
    {
        super (customerName, accountNumber);
    }

    public void withdraw (double amount)
    {
        if (balance - amount >= minimumBalance)
        {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount +
" successful");
        }
    }
}

```

11

```

else
{
    System.out.println("Insufficient funds - service charge applied");
    imposeServiceCharge();
}

private void imposeServiceCharge()
{
    double serviceCharge = 30;
    balance -= serviceCharge;
    System.out.println("Service charge of $" + serviceCharge + " imposed");
}

Output:

Deposit of $ 5000 successful
Account Number : SA1001
Balance : $ 5000.0
Interest of $ 250.0 computed and added to the balance
Account Number : SA1001
Balance : $ 5250.0
Withdrawal of $ 3000 successful
Account Number : SA1001
Balance : $ 3250.0
Deposit of $ 8000 successful
Account Number : CA2002
Balance : $ 8000.0
Withdrawal of $ 5000 successful
Account Number : CA2002
Balance : $ 3000.0

```

Q-

PROGRAM:

```
package CIE;
public class Student
{
    String name;
    String usn;
    int sem;

    public Student (String name, String usn, int sem),
    {
        this.name = name;
        this.usn = usn;
        this.sem = sem;
    }
}
```

```
package CIE;
public class Internals extends CIE.Student
{
}
```

```
public int internalMarks [];
public Internals (String name, String usn, int sem, int internal
    Marks [])
{
    Super(name, usn, sem);
    this.InternalMarks = InternalMarks;
}
```

```
package SEE;
import CIE.Student;
public class Externals extends CIE.Student
{
    public int SeeMarks [];
    public Externals (String Name, String usn, int sem, int SeeMarks[])
    {
        super (name, usn, sem);
        this.SeeMarks = SeeMarks;
    }
}
```

```
import CIE.Student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;
```

```
public class Final Marks
{
```

```
public static void main (String args [])
{
```

```
Scanner si = new Scanner (System.in);
Sopln ("In Enter the number of Students");
int n = si.nextInt();
```

```
String name [] = new String [n];
String usn [] = new String [n];
int sem [] = new int [n];
int internalMarks [][] = new int [n][5];
int SeeMarks [][] = new int [n][5];
```



for (int i=0; i < n; i++)  
{

Sopln("Enter details of student " + (i+1) + ":" );

Sopln(" Name :");

name[i] = si.next();

Sopln(" USN");

USN[i] = si.next();

Sopln(" Semester");

sem[i] = si.nextInt();

Sopln(" Enter internal Marks for 5 courses :");

for (int j=0; j < 5; j++)  
{

Sopln(" Course" + (j+1) + ":" );

InternalMarks[i][j] = si.nextInt();

Sopln(" Enter SEE marks for 5 courses");

for (int k=0; k < 5; k++)  
{

Sopln(" Course" + (k+1) + ":" );

SEEMarks[i][j] = si.nextInt();

int FinalMarks[5] = new int [n][5]

for (int i=0, i < n; i++)  
{

Internals I1 = new Internals(name[i], USN[i], sem[i],  
internalMarks[i]);

Externals E1 = new Externals(name[i], USN[i], sem[i],  
SEEMarks[i]);

11

Sopln(" Final Marks for " + n + " student in 5 courses");

for (int k=0; k < n; k++)  
{

Sopln(" name[" + k + "] : ");

Sopln(" CIE marks : " + ":" );

for (int j=0; j < 5; j++)

{

Sopln(" Course " + (j+1) + ":" + InternalMarks[k][j]);

{

Sopln("\n\n");

{

si.close();

{

{

{

Input:

Bob: 10 10 10 10 10  
10 10 10 10 10 Sem: 1

Andy: 20 20 20 20  
20 20 20 20 Sem: 2

Output:

Final marks for 2 students in 5 courses:

Bob:

CIE Marks

Course 1: 10

Course 2: 10

Course 3: 10

Course 4: 10

Course 5: 10

Andy:

CIE Marks

Course 1: 20

Course 2: 20

Course 3: 20

Course 4: 20

Course 5: 20

SEE Marks

Course 1: 10

Course 2: 10

Course 3: 10

Course 4: 10

Course 5: 10

Course 1: 20

Course 2: 20

Course 3: 90

Course 4: 20

Course 5: 20

Done by  
30/10/2024

1.

```
class WrongAge extends Exception
{
```

```
    public WrongAge (String message)
    {
        super (message);
    }
}
```

```
class Father
{
```

```
    int age;
```

```
    public Father (int age) throws WrongAge
    {
    }
```

```
    if (age < 0)
```

```
        throw new WrongAge ("Age cannot be negative");
    
```

```
    this.age = age;
}
```

```
    public int getAge()
    {
    }
```

```
    return age;
}
```

```
}
```



```
class Son extends Father
{
```

```
    int sonAge;

```

```
    public Son (int fatherAge, int sonAge) throws WrongAge
    {
    }
```

```
        super (fatherAge);
    
```

```
        if (sonAge >= fatherAge)
        {
    }
```

throw new WrongAge ("Son's age cannot be greater than or equal to father's age");  
}

```
        this.sonAge = sonAge;
    }
```

```
    public int getSonAge()
    {
    }
```

```
        return sonAge;
    }
```

```
public class main
{
```

```
    public static void main (String args[])
    {
    }
```

```
try
{
```

```
Father f = new Father (45);

```

```
Son s = new Son (45, 20);

```

```
Sopln ("Father's age : " + f.getAge ());

```

```
Sopln ("Son's age : " + s.getSonAge ());

```

```
}
catch (WrongAge e)
{

```

OUTPUT:

Father's age: 45

Son's age: 20

```
    Sopln ("Exception");
    }
```

2.

class DispMessage extends Thread

{

String msg;

int interval;

public DispMessage (String msg, int interval)

{

this. msg = msg ;

this. interval = interval ;

}

public void run()

{

while (true)

{

System.out.println(msg);

try

{

Thread.sleep (interval);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

public class main

{

public static void main (String args [] )

{

DispMessage bms Thread = new DispMessage ("BMS",  
"College of Engineering", 1000);

DispMessage cse Thread = new DispMessage ("CSE", 2000);

bms Thread.start();

cse Thread.start();

}

OUTPUT:

BMSCE

CSE

CSE

CSE

CSE

BMSCE

CSE

CSE

CSE

CSE

CSE

Done  
19/12/20

1. Creating label, button and TextField in a frame using AWT.

```

import java.awt.*;
import java.awt.event.*;

public class AWTExample extends WindowAdapter {
    frame f;
    AWTExample() {
        F = new Frame();
        F.addWindowListener(this);
        Label l = new Label("Employee Id");
        Button b = new Button("Submit");
        Textfield T = new Textfield();
        l.setBounds(20, 60, 80, 30);
        t.setBounds(20, 100, 80, 30);
        b.setBounds(100, 100, 80, 30);
        F.add(b);
        F.add(l);
        F.add(t);
        F.setSize(400, 300);
        F.setTitle("Employee Info");
        F.setLayout(null);
        F.setVisible(true);
    }
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
}

```

```

public static void main(String args[]) {
}

```

```

} AWT Example awt_obj = new AWTExample();
}

```

2. Create a button and add a action Listener for Mouse click

```

import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowAdapter implements
    ActionListener {
    frame f;
    Textfield TF;
    EventHandling() {
        F = new Frame();
        F.addWindowListener(this);
        TF = new Textfield();
        TF.setBounds(60, 50, 170, 20);
        Button b = new Button("clickme");
        b.setBounds(100, 120, 80, 30);
        F.add(b);
        F.setSize(300, 300);
        F.setTitle("Employee Info");
        F.setLayout(null);
        F.setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        TF.setText("Clicked");
    }
}

```

```

F.addActionListener(this);
b.addActionListener(this);
F.add(b);
F.add(TF);
F.setSize(300, 300);
F.setLayout(null);
F.setVisible(true);
}

```

— / —

```
public void actionPerformed (ActionEvent e)  
{
```

```
    tf. setText ("Welcome");
```

```
}
```

```
public void windowClosing (WindowEvent e)  
{
```

```
    System. exit (0);
```

```
}
```

```
public static void main (String args [])
```

```
{
```

```
    new EventHandling ();
```

```
}
```

```
}
```